Available online at www.sciencedirect.com

SCIENCE DIRECT

Discrete Applied Mathematics 154 (2006) 2010–2023

DISCRETE APPLIED MATHEMATICS

www.elsevier.com/locate/dam

Searching for a counterfeit coin with b-balance☆

Wen An Liu[a,*], Huan Huan Cui[b], Bing Qing Ma[a]

[a] *College of Mathematics and Information Science, Henan Normal University, Xinxiang 453007, People's Republic of China*

[b] *Department of Mathematics, Luoyang Normal University, Luoyang 471022, People's Republic of China*

Received 5 August 2005; received in revised form 17 February 2006; accepted 23 March 2006

Available online 18 May 2006

Abstract

We consider the classical problem of searching for a heavier coin in a set of n coins, $n-1$ of which have the same weight. The weighing device is b-balance which is the generalization of two-arms balance. The minimum numbers of weighings are determined exactly for worst-case sequential algorithm, average-case sequential algorithm, worst-case predetermined algorithm, average-case predetermined algorithm.

We also investigate the above search model with additional constraint: each weighing is only allowed to use the coins that are still in doubt. We present a worst-case optimal sequential algorithm and an average-case optimal sequential algorithm requiring the minimum numbers of weighings.

Keywords: Search; Counterfeit coin; b-balance; Worst-case; Average-case

1. Introduction

The problem of searching for a heavier coin in a set of n coins is a classical problem in the area of combinational search theory and has received considerable attention. The aim of this problem is to identify the heavier coin using as few weighings as possible. Many papers have been devoted to the case when the testing device is a two-arms balance (see [1,7] and references therein). There are other testing devices, such as spring scale [1, Chapter 2], multi-arms balance [2,3]. The problem of searching for more than one counterfeit coin is also very popular [4,6,9,11–14,17,18]. Usually, two classes of algorithms are considered: *sequential* and *predetermined* algorithms. Moreover, the goodness of an algorithm is estimated by two measures: the *worst-case* number of weighings and the *average-case* number of weighings sufficient to find the heavier coin. In the average case, it is assumed that one is given a probability distribution $p=(p_1,\ldots,p_n)$, where p_i is the probability that the ith coin is heavier. For more details on these concepts, we refer the reader to Aigner [1].

Halbeise and Hangerbühler [5] have considered a new type of test device: *b-balance*. That is, in a single test we have b two-arms balances to use. The authors in [5] give an answer to their question for worst-case sequential and predetermined algorithms. In this paper, we consider the problem of searching for a heavier coin among a set of n coins when the test device is b-balance. In Section 3, the minimum worst-case numbers of weighings are determined

☆ This work is supported by the National Natural Science Foundations of China, Grant nos. 10571045 and 70571024.

* Corresponding author.

E-mail address: liuwenan@mail.china.com (W.A. Liu).

0166-218X/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2006.03.010

exactly for sequential and predetermined algorithms. In Section 4, the minimum average-case numbers of weighings are determined exactly for sequential and predetermined algorithms.

We also investigate the above search model with additional constraint: each weighing is only allowed to use the coins that are still in doubt. Section 5 presents a worst-case optimal sequential algorithm and an average-case optimal sequential algorithm requiring the minimum numbers of weighings.

2. Notations and preliminaries

We follow the notations of the case $b=1$, i.e., the test device is a *two-arms balance*. Let $S=\{1,2,\ldots,n\}$ be the *initial set of n coins*. $A:B$ is called a *weighing* if $A, B\subset S$, $A\cap B=\emptyset$ and $|A|=|B|$ (no information can be obtained by weighing two unequal-sized sets, see [1]). A weighing $A:B$ means that we perform the weighing of A against B and A, B is placed on the left, the right pan of the two-arms balance, respectively. The outcome of one weighing must be one of the three possible *feedbacks*: " left-heavy ", " right-heavy " or " equal ". It is obvious that feedback " left-heavy ", " right-heavy " and " equal " means that the heavier coin is contained in A, B and $S-A-B$, respectively.

We use the notation

$$A_1:B_1 \mid A_2:B_2 \mid \cdots \mid A_b:B_b \tag{1}$$

to denote a weighing in the present case, i.e., the test device is b-balance. One weighing of form Eq. (1) means that we put subsets of coins A_k and B_k on the left pan and the right pan of balance k, respectively, $1\leqslant k\leqslant b$. A weighing $A_1:B_1\mid A_2:B_2\mid\cdots\mid A_b:B_b$ is called *feasible* if $|A_k|=|B_k|$ for $k=1,2,\ldots,b$. When one feasible weighing is performed, the outcome of this weighing must be one of $2b+1$ possible feedbacks (denoted by $f\in[-b,b]\triangleq\{-1,1,-2,2,\ldots,-b,b,0\}$):

$-b\leqslant f=-k\leqslant -1$ means that the feedback of balance k is "left-heavy" and those of the others are "equal", i.e., the heavier coin is contained in A_k.

$1\leqslant f=k\leqslant b$ means that the feedback of balance k is "right-heavy" and those of the others are "equal", i.e., the heavier coin is contained in B_k.

$f=0$ means that all feedbacks of b balances are "equal", i.e., the heavier coin is not contained in $\bigcup_{k=1}^{b}(A_k\cup B_k)$.

When weighing $A_1:B_1\mid A_2:B_2\mid\cdots\mid A_b:B_b$ is performed and we receive a feedback f, a search domain being consistent with the feedback f can be determined uniquely, denoted by S^f. Generally, for any integer $\ell\geqslant 1$, $S^{f_1f_2\cdots f_\ell}$ denotes the search domain determined by the feedback sequence $f_1f_2\cdots f_\ell$ of these ℓ weighings. A search domain $S^{f_1f_2\cdots f_\ell}$ is called to be *final* if $|S^{f_1f_2\cdots f_\ell}|=1$. A sequential algorithm of the present model is called *admissible* if every weighing is feasible.

We call a tree $(2b+1)$*-ary* if each node has at most $2b+1$ sons, called f-son, $f\in[-b,b]$, respectively. A sequential algorithm of the present model can be represented by a $(2b+1)$-ary tree T whose root corresponds to the initial search domain and whose *leaves* correspond to the *final* search domains; each *internal node* corresponds to a search domain $S^{f_1f_2\cdots f_\ell}$. A $(2b+1)$-ary tree T is called *admissible* if the corresponding algorithm is admissible.

A predetermined algorithm can be represented by an $m\times n$ matrix $M=(a_{ij})$, where $a_{ij}\in[-b,b]$, $i=1,2,\ldots,m$, $j=1,2,\ldots,n$. Given an $m\times n$ matrix $M=(a_{ij})$, we choose the ith weighing $A_1^{(i)}:B_1^{(i)}\mid A_2^{(i)}:B_2^{(i)}\mid\cdots\mid A_b^{(i)}:B_b^{(i)}$ in the following way: for $1\leqslant i\leqslant m$,

$$A_k^{(i)}=\{j\,|\,a_{ij}=-k,\ j=1,2,\ldots,n\},\quad k\in\{1,2,\ldots,b\},$$
$$B_k^{(i)}=\{j\,|\,a_{ij}=k,\ j=1,2,\ldots,n\},\quad k\in\{1,2,\ldots,b\}.$$

We call a matrix $M=(a_{ij})$ *admissible* if it satisfies the following two conditions:

(I) In each row i, the number of elements k equals the number of elements $-k$ for $k\in\{1,2,\ldots,b\}$, i.e., $|A_k^{(i)}|=|B_k^{(i)}|$ for $k\in\{1,2,\ldots,b\}$.

(II) All columns of M are distinct.

If an admissible matrix M does not contain the 0-column, then M is said to have *the* 0*-property*. By B_L we denote the matrix consisting of all distinct columns of length L over $[-b,b]$, then B_L is an $L\times(2b+1)^L$ matrix and is obviously admissible.

3. The minimum worst-case numbers of weighings

In this section we will determine the minimum number $L(n)$ of weighings of the worst-case sequential algorithm, and the minimum number $L_{\text{pre}}(n)$ of weighings of the worst-case predetermined algorithm. It is obvious that

$$\lceil \log_{2b+1} n \rceil \leqslant L(n) \leqslant L_{\text{pre}}(n). \tag{2}$$

Theorem 1. *For* $|S| = n \geqslant 2$,

$$L(n) = L_{\text{pre}}(n) = \lceil \log_{2b+1} n \rceil.$$

Furthermore, let $P = \{(2b+1)^L, (2b+1)^L - 2 | L \geqslant 2\}$. *For* $2 \leqslant n \leqslant 2b+1$ *even or* $n \notin P$, *the admissible matrix can be chosen so that it does not contain the* 0*-column*; *For* $2 \leqslant n \leqslant 2b+1$ *odd or* $n \in P$, *the* 0*-column must appear.*

Proof. It suffices to show that $L_{\text{pre}}(n) = \lceil \log_{2b+1} n \rceil$ by Eq. (2). Let n be any integer with $(2b+1)^{L-1} < n \leqslant (2b+1)^L$ and set $n = (2b+1)^{L-1} + r$, $1 \leqslant r \leqslant 2b(2b+1)^{L-1}$. It is enough to construct an admissible $L \times n$ matrix. We proceed by induction on L. For $L = 1$, we have $2 \leqslant n \leqslant 2b+1$. For n even, the desired admissible matrix with the 0-property can be constructed as follows:

$$\left(-1, 1, -2, 2, \ldots, -\frac{n}{2}, \frac{n}{2}\right).$$

For n odd, the desired admissible matrix can be constructed as follows:

$$\left(-1, 1, \ldots, -\frac{n-1}{2}, \frac{n-1}{2}, 0\right).$$

Let $L \geqslant 2$. If $n = (2b+1)^L$, then B_L is clearly the only admissible $L \times n$ matrix (containing the 0-column). If $n = (2b+1)^L - 2$, then the only way to obtain an admissible $L \times n$ matrix M is to remove a pair of distinct columns $\underline{a}, -\underline{a}$ from B_L. Hence, M again contains the 0-column, thus proving our last claim. Let us now assume $n \notin P = \{(2b+1)^L, (2b+1)^L - 2 | L \geqslant 2\}$.

If n is even, then we remove $((2b+1)^L - 1 - n)/2$ pairs $\underline{a}, -\underline{a}$ and the 0-column from B_L, thus obtaining an admissible search matrix of size $L \times n$. If n is odd, then $n \leqslant (2b+1)^L - 4$. In this case we first remove from B_L the 0-column and the three columns

$$\begin{matrix} b & 0 & -b \\ 0 & b & -b \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0. \end{matrix}$$

Since $n > (2b+1)^{L-1} \geqslant (2b+1) \geqslant 3$, we have at least $((2b+1)^L - 4 - n)/2$ pairs of columns $\underline{a}, -\underline{a}$ in the remaining matrix. Removing $((2b+1)^L - 4 - n)/2$ such pairs yields again an admissible matrix of size $L \times n$. $\square$

4. The minimum average numbers of weighings

Let us turn to the average case when the uniform distribution is assumed. We will determine the minimum average number $\bar{L}(n)$ of weighings for the sequential algorithm, and the minimum average number $\bar{L}_{\text{pre}}(n)$ of weighings for the predetermined algorithm.

Given a $(q+1)$-ary tree T with n leaves, let $\ell(i, T)$ be the length of the leaf i in T, i.e., the distance of i from the root of T. The *external path length* of T is defined by $h(T) = \sum_{i=1}^{n} \ell(i, T)$. Let $H(n) = \min\{h(T)\}$, where the minimum is taken over all $(q+1)$-ary trees with n leaves. We call tree T^* *Huffman tree* if $h(T^*) = H(n)$. Determining the quantity $H(n)$ and obtaining the structure of Huffman tree T^* is called the *Huffman problem*. The following Lemma 2 is the solution to Huffman problem. By $\lfloor x \rfloor$ and $\lceil x \rceil$ we denote the maximal integer $\leqslant x$ and the minimal integer $\geqslant x$, respectively.

Lemma 2. *Given an integer n with* $(q+1)^L \leqslant n < (q+1)^{L+1}$. *Let* $n = (q+1)^L + qk + t$, *where* $0 \leqslant k < (q+1)^L$, $0 \leqslant t \leqslant q-1$. T^* *is a Huffman tree if and only if* T^* *has* $\lambda_1 = n - \lceil (qk+t)(q+1)/q \rceil$ *leaves at level* $L = \lfloor \log_{q+1} n \rfloor$ *and* $\lambda_2 = \lceil (qk+t)(q+1)/q \rceil$ *leaves at level* $L+1$. *Moreover,*

$$H(n) = \lambda_1 L + \lambda_2 (L+1) = n \lfloor \log_{q+1} n \rfloor + \lceil (qk+t)(q+1)/q \rceil. \tag{3}$$

Let T_L *be the tree with* $(q+1)^L$ *leaves on level L. A Huffman tree with n leaves can be obtained from* T_L *by changing k leaves into internal nodes each having* $q+1$ *sons if* $t=0$, *and one more leaf into internal node having* $t+1$ *sons if* $t>0$.

Proof. See [1, 2]. □

By $\mathscr{T}_{\mathrm{adm}}(n)$ we denote the class of all $(2b+1)$-ary admissible trees with n leaves, and let

$$e(n) = \min\{h(T) | T \in \mathscr{T}_{\mathrm{adm}}(n)\}. \tag{4}$$

It is obvious that $\bar{L}(n) = \min\{h(T)/n | T \in \mathscr{T}_{\mathrm{adm}}(n)\} = e(n)/n$. Essentially, determining the quantity $e(n)$ is a special case of the Huffman problem because we are required to obtain a restricted $(2b+1)$-ary tree (*admissible* tree) T with $h(T) = e(n)$, where the restriction is required by the test device (b-balance). It follows from Lemma 2 that $H(n)$ is a lower bound of $e(n)$, i.e., $e(n) \geqslant H(n)$ for all integers $n \geqslant 2$. Thus,

$$\frac{H(n)}{n} \leqslant \bar{L}(n) \leqslant \bar{L}_{\mathrm{pre}}(n). \tag{5}$$

For easy citation, we now give an equivalent formula of $H(n)$. Given an integer n with $(2b+1)^L \leqslant n < (2b+1)^{L+1}$. Let $n = (2b+1)^L + 2bk + t$, where $0 \leqslant k < (2b+1)^L$, $0 \leqslant t \leqslant 2b-1$, and let

$$\alpha(t) = \begin{cases} 0 & \text{if } t = 0, \\ 1 & \text{if } t > 0. \end{cases}$$

By letting $q = 2b$ in Eq. (3), we have

$$H(n) = nL + k(2b+1) + t + \alpha(t). \tag{6}$$

A predetermined algorithm $\mathscr{A}$ sufficient to identify the heavier coin among a set of n coins can be represented by an admissible $m \times n$ matrix M, where the n columns are of the same length m, that is, the worst-case number of weighings. Suppose the first ℓ_j weighings in $\mathscr{A}$ can uniquely identify the heavier coin when the heavier coin is the jth one, then $m = \max_{1 \leqslant j \leqslant n}\{\ell_j\}$. The heavier coin is the jth one only if the feedback to the m weighings is just the jth column. Since the jth coin is uniquely identified after ℓ_j weighings, we can cross out the last $m - \ell_j$ entries in the jth column of M. As a consequence, a predetermined algorithm in the average case can be represented by a table which may have columns of different length. If a table representing the algorithm contains no pair of columns such that one is the prefix of the other, then the heavier coin is uniquely identified as soon as the sequence of feedbacks coincides with a column of M. The index of this column corresponds to the index of the heavier coin. For more details, see Bonis [2].

So an n-column table M with elements $a_{ij} \in [-b, b]$ represents a predetermined algorithm in the average case if and only if the following two conditions hold:

(I) In each row, the number of elements k equals the number of elements $-k$ for $k \in \{1, 2, \ldots, b\}$.

(II′) For each pair M_i and M_j of columns of M, M_i is not a prefix of M_j.

An n-column table M which satisfies the conditions (I) and (II′) is said *admissible*. Let $N(M)$ be the number of non-empty entries of an n-column admissible table M, and let $N(n) = \min\{N(M)\}$, where the minimum is taken over all n-column admissible table M. Then the minimum average number $\bar{L}_{\mathrm{pre}}(n) = N(n)/n$. Lemma 2 shows that $N(n) \geqslant H(n)$, thus $\bar{L}_{\mathrm{pre}}(n) \geqslant H(n)/n$ for all integers $n \geqslant 2$.

The following Theorem 3 gives the minimum average number $\bar{L}_{\mathrm{pre}}(n)$ of weighings for predetermined algorithm. To simplify the proof of Theorem 3, we establish the following notations:

- For integers i, ℓ and m, we denote by $i^{\ell\times m}$ the table with ℓ rows and m columns having each entry equal to i.
- Given two integers i and j with $i \leqslant j$, the notation $i \cdots j$ indicates $j-i+1$ consecutive row entries containing the integers from i to j.
- If $M_1=(a_{ij})$ is an $\ell \times m_1$ matrix and $M_2=(b_{ij})$ is an $\ell \times m_2$ matrix, we use $(M_1|M_2)$ to denote the new $\ell \times (m_1+m_2)$ matrix by appending M_2 to M_1. Let $\sum_{i=1}^{k} M_i = (M_1|M_2|\cdots|M_k)$, and $k \cdot M = \sum_{i=1}^{k} M$. If v is a $1 \times m_1$ matrix, we denote by $\begin{bmatrix} M_1 \\ v \end{bmatrix}$ the new $(\ell+1) \times m_1$ matrix by appending a last row v to M_1.

Theorem 3. *Given an integer $n \geqslant 2$. Suppose $n=(2b+1)^L+2bk+t$, where $L \geqslant 0$, $0 \leqslant k < (2b+1)^L$, $0 \leqslant t < 2b$, i.e., $(2b+1)^L \leqslant n < (2b+1)^{L+1}$. Then*

$$\bar{L}_{\mathrm{pre}}(n) = \begin{cases} \dfrac{H(n)+1}{n} & \text{if } (L=1,\ k \text{ odd},\ t \text{ odd}) \\ & \text{or } (L>1,\ k=1,\ t \text{ odd}) \\ & \text{or } (L>1,\ k=(2b+1)^L-2,\ t=2b-1), \\ \dfrac{H(n)}{n} & \text{otherwise}. \end{cases}$$

Proof. We note that $n=(2b+1)^L+2bk+t$, where $L \geqslant 0$, $0 \leqslant k < (2b+1)^L$, $0 \leqslant t < 2b$, i.e., $(2b+1)^L \leqslant n < (2b+1)^{L+1}$. We will show that there exists an admissible n-column table M such that $N(M)=H(n)$ for some cases, thus $\bar{L}_{\mathrm{pre}}(n)=H(n)/n$. By Lemma 2, it is enough to construct an admissible n-column table M which has $\lambda_1=n-\lceil(2bk+t)(2b+1)/(2b)\rceil$ columns of length $\lfloor \log_{2b+1} n \rfloor = L$ and $\lambda_2=\lceil(2bk+t)(2b+1)/(2b)\rceil$ columns of length $L+1$. We proceed by induction on L. We note that $\lambda_2=(2b+1)k+t+\alpha(t)$ and $\lambda_1=n-\lambda_2$.

For $L=0$, we have $2 \leqslant n < 2b+1$, $\lambda_1=0$ and $\lambda_2=n$. The desired admissible table exists by Theorem 1.

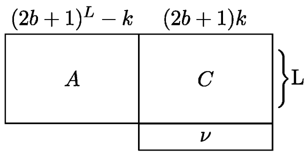
Case 1: $L \geqslant 1$, $k=0$ and $t=0$. Then $n=(2b+1)^L$, $\lambda_1=n$ and $\lambda_2=0$. The only admissible $L \times (2b+1)^L$ matrix B_L is our desired table.

Case 2: $L \geqslant 1$, $k=0$ and $1 \leqslant t < 2b$. Then $\lambda_1=(2b+1)^L-1$, $\lambda_2=t+1$. By $\underline{0}$ we denote the 0-column $0^{L\times 1}$. We construct the admissible table M with $N(M)=H(n)$ as follows

$$M=(B_L-\underline{0})+\begin{bmatrix} 0^{L\times(t+1)} \\ v \end{bmatrix},$$

where v is an admissible $1 \times (t+1)$ table which exists by Theorem 1.

Case 3: $L \geqslant 1$, $t=0$ and $1 \leqslant k < (2b+1)^L$. Then $\lambda_1=(2b+1)^L-k$, $\lambda_2=(2b+1)k$. Decompose M as in the figure

$M=$ 

By the prefix property all columns of A are distinct and the columns of A do not appear in C. Hence any column $\underline{c}$ in C appears precisely $(2b+1)$ times and is bordered by $-b, -b+1, \ldots, b$. In other words, if we assume $\underline{c}_1, \ldots, \underline{c}_k$ be the distinct columns of C (each appearing $(2b+1)$ times) and matrix $C'=\sum_{i=1}^{k} \underline{c}_i$, then

$$\begin{bmatrix} C \\ v \end{bmatrix} = \sum_{i=1}^{k} \begin{bmatrix} (2b+1)\cdot \underline{c}_i \\ -b \cdots b \end{bmatrix}.$$

We note that $(A|C)=B_L+2b\cdot C'$. It is easily seen that M satisfies (I) only if the matrix C' satisfies (I). But the converse is also true: If we can find k vectors of length L which make up a matrix C' with property (I), then make them each appear $(2b+1)$ times, bordering each of them by $-b, -b+1, \ldots, b$ and adding the other $(2b+1)^L-k$ columns of length L $(A=B_L-C')$, we clearly obtain an admissible table M with $N(M)=H(n)$. So our problem reduces to finding k such distinct vectors of length L. This latter problem is, however, immediately settled by just invoking Theorem 1. If

$(2b+1)^{s-1} < k \leqslant (2b+1)^s$, $s \leqslant L$, we take an admissible $s \times k$ matrix and add $(L-s)$ 0-rows at the bottom. If $k=1$, then we choose the 0-column.

Case 4: ($L=1$, $1 \leqslant k < 2b+1$ even, $t>0$) or ($L>1$, $1 < k \neq (2b+1)^L - 2$, $t>0$). Now $\lambda_1 = (2b+1)^L - (k+1)$, $\lambda_2 = (2b+1)k + t + 1$. We decompose M in this case as follows:

$$M = \begin{array}{c|c|c|l} \multicolumn{1}{c}{} & \multicolumn{1}{c}{(2b+1)^L-(k+1)} & \multicolumn{1}{c}{(2b+1)k+t+1} & \\ \cline{2-3} & A & C & \}L \\ \cline{2-3} \multicolumn{2}{c|}{} & \nu & \\ \cline{3-3} \end{array}$$

By the same reasoning as in case 3, it follows that C contains $k+1$ distinct columns (all distinct from the columns of A). It might be that k of them appear $(2b+1)$ times whereas one column appear $(t+1)$ times. Let these columns be $\underline{c}_1, \ldots, \underline{c}_k$ and $\underline{c}$. Each column $\underline{c}_i$ must be bordered by $-b, -b+1, \ldots, b$. Whereas $\underline{c}$ must be bordered by an admissible $1 \times (t+1)$ matrix v'. If we set $\underline{c} = \underline{0}$ and the $\underline{c}_i$'s make up an admissible $L \times k$ matrix with $\underline{c}_i \neq \underline{0}$, then M is admissible. So we have to find an admissible $L \times k$ matrix with non-zero columns. This latter problem is, however, immediately settled by just invoking Theorem 1. If $(2b+1)^{s-1} < k \leqslant (2b+1)^s$ with $1 \leqslant s < L$, then we take any admissible $s \times k$ matrix S, add $1, -1$ if there is a 0-column in S, and fill up with 0's.

$$\begin{array}{r|l|} \cline{2-2} & 0 & \\ s\left\{\begin{array}{c} \\ \\ \end{array}\right. & \vdots \quad S & \\ & 0 & \\ \cline{2-2} & 1\ -1\ 0 \cdots\cdots 0 & \\ \cline{2-2} L-1-s & 0 \cdots \ \cdots \ \cdots 0 & \\ \cline{2-2} \end{array}$$

Case 5: $L \geqslant 1$, k odd, $t>0$ even. Let $D = \{\underline{a}_1, -\underline{a}_1, \ldots, \underline{a}_{(k+1)/2}, -\underline{a}_{(k+1)/2}\}$. That is, D is the admissible $L \times (k+1)$ table consisting of distinct columns $\underline{a}_1, -\underline{a}_1, \ldots, \underline{a}_{(k+1)/2}, -\underline{a}_{(k+1)/2}$ of B_L, $D' = D - \{\underline{a}_1, -\underline{a}_1\}$. We can construct the admissible table

$$M = (B_L - D) + \sum_{i=-b}^{b} \begin{bmatrix} D' \\ i^{1\times(k-1)} \end{bmatrix} + \sum_{i=1}^{b} \left(\begin{bmatrix} \underline{a}_1 \\ i \end{bmatrix} + \begin{bmatrix} -\underline{a}_1 \\ -i \end{bmatrix} \right) + \sum_{i=-(1+\frac{t}{2})}^{-1} \left(\begin{bmatrix} \underline{a}_1 \\ i \end{bmatrix} + \begin{bmatrix} -\underline{a}_1 \\ -i \end{bmatrix} \right).$$

It is easy to check that M is an admissible n-column table, the number of columns of length $L+1$ is $(2b+1)(k-1) + 2b + (2+t) = (2b+1)k + t + 1 = \lambda_2 = n - ((2b+1)^L - k - 1)$, the number of columns of length L is $(2b+1)^L - (k+1) = n - \lambda_2 = \lambda_1$.

Case 6: $L>1$, $k = (2b+1)^L - 2$ and $0 < t \leqslant 2b-3$ odd. In this case, we have $b>1$. Let $s = 2b - t$, then $3 \leqslant s \leqslant 2b-1$ odd. By Theorem 1, there exists an admissible $L \times (k+1-s)$ matrix D which has the 0-property as $(2b+1)^{L-1} < k+1-s = (2b+1)^L - (2b+1) + t \leqslant (2b+1)^L - 4$. We construct the admissible table as follows:

$$M = 0^{L\times 1} + \sum_{i=1}^{b} \begin{bmatrix} B_L - \underline{0} \\ i^{1\times(k+1)} \end{bmatrix} + \sum_{i=-b}^{-1} \begin{bmatrix} B_L - \underline{0} \\ i^{1\times(k+1)} \end{bmatrix} + \begin{bmatrix} D \\ 0^{1\times(k+1-s)} \end{bmatrix}.$$

It is easy to check that M is an admissible n-column table, the number of columns of length $L+1$ is $2b(k+1) + (k+1-s) = (2b+1)k + t + 1 = \lambda_2 = n - 1$, the number of columns of length L is $1 = n - \lambda_2 = \lambda_1$.

The remaining cases are $L=1$, $1 \leqslant k < 2b+1$ odd, t odd; $L>1$, $k=1$, t odd; and $L>1$, $k = (2b+1)^L - 2$, $t = 2b-1$. For these cases, it is impossible to construct an admissible table M with $N(M) = H(n)$. In fact, for the first case, the proof of Theorem 6 shows that the minimum average number of the sequential algorithm $\bar{L}(n) > H(n)/n$, so the minimum average number of the predetermined algorithm $\bar{L}_{\text{pre}}(n) > H(n)/n$ as $\bar{L}_{\text{pre}}(n) \geqslant \bar{L}(n)$.

For the second case, we have $\lambda_1 = (2b+1)^L - 2$, $\lambda_2 = 2b + 1 + t + 1$. So in the decomposition of M in case 4, we conclude that there are two distinct columns in C. Suppose the two columns are $\underline{c}$, $\underline{c'}$ and they appear $x+1$ and $y+1$ times in C, respectively, then $x + y = 2b + t$. To ensure that M is an admissible table,

$$(A|C) = (B_L | x \cdot \underline{c} | y \cdot \underline{c'})$$

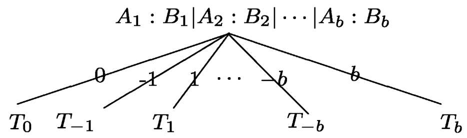


Fig. 1. General $(2b+1)$-ary admissible tree.

must satisfy (I). Since B_L satisfies (I), then $(x \cdot \underline{c}|y \cdot \underline{c'})$ has to satisfy (I). For $1 \leqslant j \leqslant L$, if the element of the *j*th row of $\underline{c}$ is $i \in [-b, b]$, then the element of the *j*th row of $\underline{c'}$ must be $-i$. So $\underline{c}$ and $\underline{c'}$ should be the negative of each other. Furthermore, to satisfy (I) we must have $x = y$. But $x + y = 2b + t$ is odd. So it is impossible to construct an admissible table M with $N(M) = H(n)$.

For the third case, we have $\lambda_1 = 1$, $\lambda_2 = (2b+1)[(2b+1)^L - 2] + 2b$. This means that A contains only one column in the decomposition of case 4. We suppose it to be $\underline{a}$. By the prefix property, the matrix C' consisting of the distinct columns of C is $B_L - \{\underline{a}\}$ and the number of columns of C' is $(2b+1)^L - 1$, $(2b+1)^L - 2$ of which appear $2b+1$ times whereas one column (denoted by $\underline{a'}$) appears $2b$, times. To ensure that the last row satisfies (I), each column of $B_L - \{\underline{a}, \underline{a'}\}$ must be bordered by $-b, -b+1, \ldots, -1, 0, 1, 2, \ldots, b$ whereas $\underline{a'}$ must be bordered by $-b, -b+1, \ldots, -1, 1, 2, \ldots, b$.

For any $m \times n$ matrix $D = (d_{ij})$, let $u_i = \sum_{j=1}^{n} d_{ij}$, $i = 1, 2, \ldots, m$. We call *rowsum*$(D) = (u_1, u_2, \ldots, u_m)^{\mathrm{T}}$ the *row-sum vector* of a matrix D (it is an $m \times 1$ matrix). Since M satisfies (I), we must have

$$\begin{aligned}\underline{0} &= \mathit{rowsum}(A|C) = \mathit{rowsum}(\underline{a} + (2b+1)(B_L - \underline{a} - \underline{a'}) + 2b\underline{a'}) \\ &= (2b+1)\mathit{rowsum}(B_L) - 2b\underline{a} - \underline{a'} \\ &= -(2b\underline{a} + \underline{a'}).\end{aligned}$$

Thus, $\underline{a'} = -2b\underline{a}$, from which we get $\underline{a'} = \underline{a} = \underline{0}$. So it is impossible to construct an admissible table M with $N(M) = H(n)$.

To finish the proof, it suffices to show that there exists an admissible table M with $N(M) = H(n) + 1$ for the above three cases, respectively. Let

$$M = (B_L - \{\underline{0}, \underline{a}, -\underline{a}\} - D) + \sum_{i=-b}^{b} \begin{bmatrix} D \\ i^{1\times(k-1)} \end{bmatrix} + \sum_{i=-(t+1)/2}^{b} \left(\begin{bmatrix} \underline{a} \\ i \end{bmatrix} + \begin{bmatrix} -\underline{a} \\ -i \end{bmatrix} \right),$$

where $\underline{a}$ is an arbitrary non-zero column of B_L, and D is an admissible $L \times (k-1)$ matrix which is obtained by deleting $((2b+1)^L - 3 - (k-1))/2$ pairs of columns $\underline{c}, -\underline{c}$ from $B_L - \{\underline{0}, \underline{a}, -\underline{a}\}$ (note that $k - 1 \leqslant (2b+1)^L - 3$ and $(2b+1)^L - 3 - (k-1)$ is even). It is easy to check that M is an admissible n-column table, the number of columns of length $L+1$ is $(2b+1)(k-1) + (2b+2+t+1) = (2b+1)k + t + 2 = \lambda_2 + 1$, the number of columns of length L is $(2b+1)^L - 3 - (k-1) = n - \lambda_2 - 1 = \lambda_1 - 1$. Thus, $N(M) = (\lambda_1 - 1) \cdot L + (\lambda_2 + 1) \cdot (L+1) = (\lambda_1 + \lambda_2) \cdot L + \lambda_2 + 1 = H(n) + 1$. $\square$

We now determine the minimum average number $\bar{L}(n)$ of weighings for sequential algorithm. Suppose that $S = \{1, 2, \ldots, n\}$ is the set of n coins and we choose $A_1 : B_1 \mid A_2 : B_2 \mid \cdots \mid A_b : B_b$ as the first weighing, where $|A_i| = |B_i|$ for $1 \leqslant i \leqslant b$. Let $C_0 = S - \bigcup_{i=1}^{b}(A_i \cup B_i)$. An admissible $(2b+1)$-ary tree T is shown in Fig. 1, where T_f denotes the subtree of T rooted at the *f*-son of T. By $|T|$ we denote the number of leaves of T. It is easy to see that $|T_{-i}| = |A_i|$, $|T_i| = |B_i|$ for $1 \leqslant i \leqslant b$, $|T_0| = |C_0|$ and

$$h(T) = n + \sum_{i=1}^{b} (h(T_{-i}) + h(T_i)) + h(T_0). \tag{7}$$

In order to prove Theorem 6, we need the following lemmas.

Lemma 4. *If $2 \leqslant n \leqslant 2b$, then there exists an admissible tree T with $h(T) = H(n) = n$.*

Proof. For $2 \leqslant n \leqslant 2b$, we can represent $n = (2b+1)^L + 2bk + t$, where $L = 0$, $k = 0$ and $1 \leqslant t \leqslant 2b - 1$, i.e., $t = n - 1$. By Eq. (6), we know $H(n) = t + 1 = n$. We choose $A_1 : B_1 \mid A_2 : B_2 \mid \cdots \mid A_b : B_b$ as the first weighing, where $|A_i| = |B_i| = 1$

for $1 \leqslant i \leqslant \lfloor n/2 \rfloor$, and $|A_i| = |B_i| = 0$ for $\lfloor n/2 \rfloor + 1 \leqslant i \leqslant b$. We note that $|C_0| = n - \sum_{i=1}^{b}(|A_i| + |B_i|) = n - 2\lfloor n/2 \rfloor = 0$ or 1 if n is even or odd, respectively. It follows from Eq. (7) that $h(T) = n = H(n)$. $\square$

Lemma 5. *For $n = (2b+1)^L$, there exists an admissible tree $T^{(L)}$ with $h(T^{(L)}) = H(n) = (2b+1)^L \times L$.*

Proof. We proceed by induction on L. $L = 0$ is trivial. For $L = 1$, we choose $A_1 : B_1 \mid A_2 : B_2 \mid \cdots \mid A_b : B_b$ as the first weighing, where $|A_i| = |B_i| = 1$ for $1 \leqslant i \leqslant b$. Let $T^{(1)}$ be the corresponding tree, it is easy to see that $h(T^{(1)}) = 2b + 1 = H(n)$. For $n = (2b+1)^L$, let $T^{(L)}$ be the corresponding tree. We choose $|A_i| = |B_i| = (2b+1)^{L-1}$ for $1 \leqslant i \leqslant b$. By the induction hypothesis, there are algorithms such that $h(T_{-i}^{(L)}) = h(T_i^{(L)}) = h(T_0^{(L)}) = H((2b+1)^{L-1}) = (2b+1)^{L-1} \times (L-1)$ for $1 \leqslant i \leqslant b$. It follows from Eq. (7) that $h(T^{(L)}) = (2b+1)^L \times L = H(n)$. $\square$

Theorem 6. *Given an integer $n \geqslant 2$. Suppose $n = (2b+1)^L + 2bk + t$, where $L \geqslant 0$, $0 \leqslant k < (2b+1)^L$, $0 \leqslant t < 2b$ i.e., $(2b+1)^L \leqslant n < (2b+1)^{L+1}$. Then*

$$\bar{L}(n) = \begin{cases} \dfrac{H(n)+1}{n} & \text{if } L = 1,\ k \text{ odd},\ t \text{ odd}, \\[2ex] \dfrac{H(n)}{n} & \text{otherwise}. \end{cases}$$

Proof. In view of $H(n)/n \leqslant \bar{L}(n) \leqslant \bar{L}_{\mathrm{pre}}(n)$ and Theorem 3, it remains to verify the following three facts:

Fact 1. $\bar{L}(n) = H(n)/n$ for $L > 1$, $k = 1$, t odd.

Fact 2. $\bar{L}(n) = H(n)/n$ for $L > 1$, $k = (2b+1)^L - 2$, $t = 2b - 1$.

Fact 3. $\bar{L}(n) = (H(n)+1)/n$ for $L = 1$, k odd, t odd.

Proof of Fact 1. $L > 1, k = 1, t$ odd. It is obvious that $\bar{L}(n) = e(n)/n \geqslant H(n)/n$. To prove $\bar{L}(n) \leqslant H(n)/n$, it is enough to show that there exists a $(2b+1)$-ary admissible tree T with external path length $h(T) = H(n) = nL + (2b+1) + t + 1 = nL + 2b + t + 2$. We proceed by induction on L.

For $L = 2$, $n = (2b+1)^2 + 2b + t$. We choose $A_1 : B_1 \mid A_2 : B_2 \mid \cdots \mid A_b : B_b$ as the first weighing of T, where $|A_i| = |B_i| = 2b + 1$ for $1 \leqslant i \leqslant b$. Then $|C_0| = n - 2b(2b+1) = 2b + 1 + 2b + t$. By Lemma 5, there are algorithms such that $h(T_{-i}) = h(T_i) = H(2b+1) = 2b + 1$ for $1 \leqslant i \leqslant b$. It follows from Eq. (7) that

$$h(T) = n + 2b(2b+1) + h(T_0). \tag{8}$$

If the feedback of this weighing is $f = 0$, then there is a coin x^* known to be good (A_1 is now a set of good coins). We choose $A'_1 : B'_1 \mid A'_2 : B'_2 \mid \cdots \mid A'_b : B'_b \cup \{x^*\}$ as the weighing of T_0, where $|A'_i| = |B'_i| = 1$ for $1 \leqslant i \leqslant b - 1$ and $|A'_b| = (2b + t + 3)/2$, $|B'_b| = (2b + t + 1)/2$. Then $|C'_0| = |C_0 - \bigcup_{i=1}^{b}(A'_i \cup B'_i)| = 1$. We note that $2 \leqslant (2b + t + 1)/2 < (2b + t + 3)/2 \leqslant 2b + 1$. It follows from Eq. (7), Lemmas 4 and 5 that

$$h(T_0) = |C_0| + \frac{2b+t+1}{2} + \frac{2b+t+3}{2}. \tag{9}$$

Combining Eqs. (8) and (9), we have $h(T) = n \times 2 + (2b+1) + t + 1 = H(n)$ in view of Eq. (6).

For $L \geqslant 3$, we choose $A_1 : B_1 \mid A_2 : B_2 \mid \cdots \mid A_b : B_b$ as the first weighing of T, where $|A_i| = |B_i| = (2b+1)^{L-1}$ for $1 \leqslant i \leqslant b$. Then $|C_0| = n - 2b(2b+1)^{L-1} = (2b+1)^{L-1} + 2b + t$ and the induction hypothesis implies $h(T_0) = [n - 2b(2b+1)^{L-1}](L-1) + (2b+1) + t + 1$. Lemma 5 implies there are $2b$ admissible subtrees $T_{-1}, T_1, \ldots, T_{-b}, T_b$ such that $h(T_1) = h(T_{-1}) = \cdots = h(T_b) = h(T_{-b}) = (2b+1)^{L-1}(L-1)$. By Eq. (7),

$$\begin{aligned} h(T) &= n + 2b(2b+1)^{L-1}(L-1) + [n - 2b(2b+1)^{L-1}](L-1) + (2b+1) + t + 1 \\ &= nL + (2b+1) + t + 1 \\ &= H(n). \end{aligned}$$

Proof of Fact 2. $L>1$, $k=(2b+1)^L-2$, $t=2b-1$. Now $n=(2b+1)^{L+1}-(2b+1)$. The proof is similar to the proof of Fact 1. For $L=2$, we choose $A_1:B_1 \mid A_2:B_2 \mid \cdots \mid A_b:B_b$ as the first weighing of *T*, where $|A_i|=|B_i|=(2b+1)^L$ for $1\leqslant i\leqslant b$; and choose $A_1':B_1' \mid A_2':B_2' \mid \cdots \mid A_b':B_b'\cup\{x^*\}$ as the weighing of T_0, where $|A_i'|=|B_i'|=2b+1$ for $1\leqslant i\leqslant b-1$ and $|A_b'|=2b+1$, $|B_b'|=2b$. It is easy to check $h(T)=H(n)$. For $L\geqslant 3$, choosing $|A_i|=|B_i|=(2b+1)^L$ for $1\leqslant i\leqslant b$, and applying the induction hypothesis as before.

Proof of Fact 3. $L=1$, $1\leqslant k<2b+1$ odd, *t* odd. Now $n=(2b+1)+2bk+t$ is even and $\lambda_2=(2b+1)k+t+1$, $\lambda_1=n-\lambda_2=2b-k$ are odd. We will show that $h(T)>H(n)$ for any admissible tree *T* in the present case.

Suppose *T* is an admissible tree with $h(T)=H(n)$. Without loss of generality, let $A_1:B_1 \mid A_2:B_2 \mid \cdots \mid A_b:B_b$ be the first weighing of tree *T*. $|A_i|=|B_i|$ implies that either both T_{-i} and T_i are leaves at level 1 of T ($|A_i|=|B_i|=1$), or both T_{-i} and T_i are not leaves at level 1 of T ($|A_i|=|B_i|\neq 1$), i.e., the total number of leaves of T_f, for $f\in\{-i,i|1\leqslant i\leqslant b\}$ at level $L=1$ is even.

If $\sum_{i=1}^{b}(|A_i|+|B_i|)=n$, then $|C_0|=0$, i.e., T_0 is not a leaf at level 1 of *T*; If $\sum_{i=1}^{b}(|A_i|+|B_i|)\neq n$, then $|C_0|=n-\sum_{i=1}^{b}(|A_i|+|B_i|)\neq 0$. We note that $|C_0|$ is even and $|C_0|\geqslant 2$ as *n* is even, i.e., T_0 is not a leaf at level 1 of *T*.

So, we can conclude that the total number of leaves at level $L=1$ of *T* must be even. This is a contradiction to the fact that *T* has λ_1 leaves at level $L=1$ and λ_1 is odd.

Theorem 3 shows that there exists an admissible table *M* with $N(M)=H(n)+1$. An admissible table *M* is a special admissible sequential algorithm, so there exists an admissible tree *T* with $h(T)=H(n)+1$. Thus, $\bar{L}(n)=e(n)/n\leq (H(n)+1/n)$. Therefore, $\bar{L}(n)=(H(n)+1)/n$ in this case. The proof of Theorem 6 is complete. □

5. Optimal sequential algorithms on the constrained model

In this section we focus on the *Constrained Model* (**CM**): there is a heavier coin in a set of *n* coins, $n-1$ of which are good coins having the same weight. The test device is *b*-balance and we are only allowed to use coins that are still in doubt. We will determine the minimum worst-case number $L_{\mathrm{CM}}(n)$ of weighings of the sequential algorithm, and the minimum average number $\bar{L}_{\mathrm{CM}}(n)$ of the sequential algorithm when the uniform distribution is assumed.

A sequential algorithm of **CM** is called *CM-admissible* if the weighing $A_1:B_1 \mid A_2:B_2 \mid \cdots \mid A_b:B_b$ of any search domain $S^{f_1f_2\cdots f_\ell}$ satisfy the following conditions:

(III) All sets A_i, B_i $(i=1,2,\ldots,b)$ are subsets of coins contained in the search domain $S^{f_1f_2\cdots f_\ell}$ and are pairwise disjoint;

(IV) $|A_i|=|B_i|$ for $i=1,2,\ldots,b$.

Condition (III) is required by the assumption that we are only allowed to use the coins that are still in doubt. Condition (IV) is required by the fact that no information can be obtained by weighing two unequal-sized sets in one balance.

A $(2b+1)$-*ary* tree *T* is called *CM-admissible* if the corresponding sequential algorithm is CM-admissible. By $\mathscr{T}_{\mathrm{CM-adm}}(n)$ we denote the class of all $(2b+1)$-ary CM-admissible trees with *n* leaves, and *L(T)* the worst-case length of tree *T*. Then,

$$L_{\mathrm{CM}}(n)=\min\{L(T)|T\in\mathscr{T}_{\mathrm{CM-adm}}(n)\}, \tag{10}$$

$$\bar{L}_{\mathrm{CM}}(n)=\min\left\{\frac{h(T)}{n}\,\middle|\,T\in\mathscr{T}_{\mathrm{CM-adm}}(n)\right\}. \tag{11}$$

It is obvious that $\mathscr{T}_{\mathrm{CM-adm}}(n)\subseteq\mathscr{T}_{\mathrm{adm}}(n)$ and

$$\lceil\log_{2b+1}n\rceil\leqslant L(n)\leqslant L_{\mathrm{CM}}(n), \tag{12}$$

$$\frac{H(n)}{n}\leqslant\bar{L}(n)\leqslant\bar{L}_{\mathrm{CM}}(n). \tag{13}$$

Theorem 1 shows that $L(n)=\lceil\log_{2b+1}n\rceil$. The following Theorem 7 gives a stronger result.

Theorem 7. $L_{\mathrm{CM}}(n)=\lceil\log_{2b+1}n\rceil$ *for any integer n.*

Proof. It is obvious that $\lceil \log_{2b+1} n \rceil \leqslant L_{\mathrm{CM}}(n) \leqslant L(T)$. To prove Theorem 7, it suffices to show that there exists a CM-admissible tree T such that $L(T) \leqslant \lceil \log_{2b+1} n \rceil \triangleq \ell$. We prove it by induction on ℓ. We note that $\ell = \lceil \log_{2b+1} n \rceil$ if and only if $(2b+1)^{\ell-1} < n \leqslant (2b+1)^{\ell}$.

$n = 1$ is trivial. For $\ell = 1$, i.e., $2 \leqslant n \leqslant 2b+1$, we choose $|A_i| = |B_i| = 1$ for $1 \leqslant i \leqslant \lfloor n/2 \rfloor$, and $|A_i| = |B_i| = 0$ for $\lfloor n/2 \rfloor + 1 \leqslant i \leqslant b$. $\ell = 1$ weighing is enough (see Fig. 1).

$\ell \geqslant 2$. Now $(2b+1)^{\ell-1} < n \leqslant (2b+1)^{\ell}$. We represent $n = (2b+1)m + j$ for some integers m and $0 \leqslant j \leqslant 2b$. We choose $|A_i| = |B_i| = m+1$ for $1 \leqslant i \leqslant \lfloor j/2 \rfloor$, and $|A_i| = |B_i| = m$ for $\lfloor j/2 \rfloor + 1 \leqslant i \leqslant b$. Then $|C_0| = n - \sum_{i=1}^{b}(|A_i| + |B_i|) = m + (j - 2\lfloor j/2 \rfloor)$.

We note that $m \leqslant (2b+1)^{\ell-1}$ if $j = 0$, and $m+1 \leqslant (2b+1)^{\ell-1}$ if $j \neq 0$ ($m \leqslant (2b+1)^{\ell-1} - j/(2b+1) < (2b+1)^{\ell-1}$). So we conclude that $|C_0| \leqslant (2b+1)^{\ell-1}$ and $|A_i| = |B_i| \leqslant (2b+1)^{\ell-1}$ for $1 \leqslant i \leqslant b$. The induction hypothesis implies that $L(T) \leqslant 1 + (\ell - 1) = \ell$. The proof of Theorem 7 is complete. □

We now determine the quantity $\bar{L}_{\mathrm{CM}}(n)$. In order to prove Theorem 10, we need the following lemmas.

Lemma 8. *For $1 \leqslant n \leqslant 2b$, there exists a CM-admissible tree T with $h(T) = H(n) = n$; For $n = (2b+1)^L$, there exists a CM-admissible tree $T^{(L)}$ with $h(T^{(L)}) = H(n) = (2b+1)^L \times L$.*

Proof. The proof of the first assertion is the same with that of Lemma 4. The proof of the second assertion is the same with that of Lemma 5. □

Lemma 9. *Given an integer n with $(2b+1)^L \leqslant n < (2b+1)^{L+1}$ and $L \geqslant 1$. If T is a CM-admissible Huffman tree and $|T| = n$ is even, then the number of leaves at level L of T is even.*

Proof. By Lemma 2, T being a CM-admissible Huffman tree implies that T has λ_1 leaves at level L and $\lambda_2 = n - \lambda_1$ leaves at level $L+1$ ($q = 2b$ in Lemma 2), i.e., all n leaves of T are at level L or $L+1$ of T. We will prove that λ_1 is even by induction on L.

For $L = 1$, suppose that $A_1 : B_1 \mid A_2 : B_2 \mid \cdots \mid A_b : B_b$ is the first weighing of tree T. $|A_i| = |B_i|$ implies that either both T_{-i} and T_i are leaves at level 1 of T ($|A_i| = |B_i| = 1$), or both T_{-i} and T_i are not leaves at level 1 of T ($|A_i| = |B_i| \neq 1$), i.e., the total number of leaves of subtrees T_f, for $f \in \{-i, i \mid 1 \leqslant i \leqslant b\}$ at level 1 of T is even.

If $\sum_{i=1}^{b}(|A_i|+|B_i|) = n$, then $|C_0| = n - \sum_{i=1}^{b}(|A_i|+|B_i|) = 0$, i.e., T_0 is not a leaf at level 1 of T; If $\sum_{i=1}^{b}(|A_i|+|B_i|) \neq n$, then $|C_0| = n - \sum_{i=1}^{b}(|A_i| + |B_i|) \neq 0$. In this case, $|C_0|$ is even and $|C_0| \geqslant 2$ as n is even, i.e., T_0 is not a leaf at level 1 of T. We can conclude that the number λ_1 of leaves at level $L = 1$ of T must be even.

For $L \geqslant 2$, suppose that $A_1 : B_1 \mid A_2 : B_2 \mid \cdots \mid A_b : B_b$ is the first weighing of tree T. By Lemma 2, T being a Huffman tree implies that there are $(2b+1)^L$ nodes (the total number of inner nodes and leaves) at level L of T, i.e.,

$$|T_{-i}| = |T_i| = |A_i| = |B_i| \geqslant (2b+1)^{L-1} \quad \text{for } 1 \leqslant i \leqslant b,$$

and

$$|T_0| = |C_0| \geqslant (2b+1)^{L-1}.$$

T being a Huffman tree also implies that any subtree T_f, $f \in [-b, b]$, is a Huffman tree (any subtree of a Huffman tree is also a Huffman tree, see [8]). Thus it follows from Lemma 2 that subtrees T_{-i} and T_i have the same number of leaves at level L of T. We can conclude that the total number of leaves of subtrees T_f, for $f \in \{-i, i \mid 1 \leqslant i \leqslant b\}$ at level L of T is even.

On the other hand, we note that $|C_0| = n - \sum_{i=1}^{b}(|A_i| + |B_i|)$ is even as n is even and $|A_i| = |B_i|$. We claim that $|C_0| < (2b+1)^L$. Suppose, on the contrary, that $|C_0| > (2b+1)^L$ ($|C_0| \neq (2b+1)^L$ as $|C_0|$ is even). It is obvious that $|C_0| \leqslant n < (2b+1)^{L+1}$, so we represent $|C_0| = (2b+1)^L + 2bk' + t'$, where $0 \leqslant k' < (2b+1)^L$, $0 \leqslant t' \leqslant 2b-1$, and $k' \neq 0$ or $t' \neq 0$. By Lemma 2, T_0 being a Huffman tree implies that the number of leaves at level $L+1$ of subtree T_0 is $\lceil (2bk' + t')(2b+1)/2b \rceil = (2b+1)k' + t' + \alpha(t') \neq 0$, i.e., the number of leaves at level $L+2$ of T is not zero. This is a contradiction to the fact that all n leaves of T are at level L or $L+1$ of T (T is a Huffman tree). We note that T_0 is a Huffman tree, $|T_0| = |C_0|$ is even, and $(2b+1)^{L-1} \leqslant |C_0| < (2b+1)^L$, and $L - 1 \geqslant 1$. The induction hypothesis implies that the number of leaves at level $L-1$ of T_0 is even, i.e., the number of leaves at level L of T is even.

Summing up, the total number of leaves of subtrees T_f, for $f \in [-b, b]$ at level L of T is even, i.e., λ_1 is even. □

Theorem 10. *Given an integer* $n \geqslant 2$ *with* $(2b+1)^L \leqslant n < (2b+1)^{L+1}$. *Let* $n = (2b+1)^L + 2bk + t$, *where* $0 \leqslant k < (2b+1)^L$, $0 \leqslant t < 2b$. *One has*

$$\bar{L}_{\mathrm{CM}}(n) = \begin{cases} \dfrac{H(n)+1}{n} & \textit{if both } k \textit{ and } t \textit{ are odd}, \\ \dfrac{H(n)}{n} & \textit{otherwise}. \end{cases}$$

Proof. It is enough to prove the existence of a CM-admissible tree T with n leaves having external path length

$$h(T) = \begin{cases} H(n)+1 & \text{if both } k \text{ and } t \text{ are odd}, \\ H(n) & \text{otherwise}, \end{cases} \tag{14}$$

and to show that if both k and t are odd, then any CM-admissible tree has external path length $\geqslant H(n)+1$. We proceed by induction on L.

For $L = 0$, i.e., $2 \leqslant n \leqslant 2b$. Lemma 8 implies that Eq. (14) is true.

For $L = 1$, we distinguish cases $t = 0$ and $t \neq 0$.

(1) $t = 0$. Now $n = (2b+1) + 2bk$ and $H(n) = n + k(2b+1)$ (by Eq. (6)). We choose in Fig. 1 that $|A_i| = |B_i| = 2b+1$ for $1 \leqslant i \leqslant \lfloor k/2 \rfloor$, and $|A_i| = |B_i| = 1$ for $\lfloor k/2 \rfloor + 1 \leqslant i \leqslant b$. Then $|C_0| = n - \sum_{i=1}^{b}(|A_i| + |B_i|) = 1 + 2b(k - 2\lfloor k/2 \rfloor) = 1$ or $2b+1$ if k is even or odd, respectively. Lemma 8 shows that there exists a CM-admissible tree T_0 such that $h(T_0) = 0$ or $2b+1$ if k is even or odd, respectively; and also there exist CM-admissible trees T_{-i} and T_i $(1 \leqslant i \leqslant b)$ such that $h(T_{-i}) = h(T_i) = H(2b+1) = 2b+1$ for $1 \leqslant i \leqslant \lfloor k/2 \rfloor$, and $h(T_{-i}) = h(T_i) = H(1) = 0$ for $\lfloor k/2 \rfloor + 1 \leqslant i \leqslant b$, respectively. It follows from Eqs. (6) and (7) that $h(T) = n + k(2b+1) = H(n)$.

(2) $1 \leqslant t \leqslant 2b-1$. If k is even, by taking $|A_i| = |B_i| = 2b+1$ for $1 \leqslant i \leqslant k/2$, and $|A_i| = |B_i| = 1$ for $k/2 + 1 \leqslant i \leqslant b$, we have $2 \leqslant |C_0| = t + 1 \leqslant 2b$. It follows from Lemma 8, Eqs. (6) and (7) that $h(T) = n + k(2b+1) + t + 1 = H(n)$.

If k is odd and t is even, by taking $|A_i| = |B_i| = 2b+1$ for $1 \leqslant i \leqslant (k-1)/2$, and $|A_i| = |B_i| = 1$ for $(k+1)/2 \leqslant i \leqslant b-1$, and $|A_{-b}| = |B_b| = b + 1 + t/2$, we have $|C_0| = 1$ and $2 \leqslant |A_{-b}| = |B_b| \leqslant 2b$. It follows from Lemma 8, Eqs. (6) and (7) that $h(T) = n + k(2b+1) + t + 1 = H(n)$.

If both k and t are odd, we cannot get any CM-admissible tree which has external path length $H(n)$. Suppose that there exists a CM-admissible tree T with $h(T) = H(n)$, then T has $n - \lceil (2bk+t)(2b+1)/2b \rceil = 2b - k$ leaves at level 1 (by letting $q = 2b$ in Lemma 2). On the other hand, both k and t being odd implies that $n = (2b+1) + 2bk + t$ is even. Lemma 9 shows that the number of leaves at level 1 of T is even. This is a contradiction to the fact that $2b - k$ is odd.

A CM-admissible tree T which has external path length $H(n)+1$ can be obtained below. In Fig. 1, we take $|A_i| = |B_i| = 2b+1$ for $1 \leqslant i \leqslant (k-1)/2$, and $|A_i| = |B_i| = 1$ for $(k+1)/2 \leqslant i \leqslant b-1$, and $|A_{-b}| = |B_b| = 1 + (t+1)/2$. Then $|C_0| = 2b$. Lemma 8 shows that there exist CM-admissible trees T_0, T_{-i} and T_i $(1 \leqslant i \leqslant b)$ such that $h(T_0) = H(2b) = 2b$, and $h(T_{-i}) = h(T_i) = H(2b+1) = 2b+1$ for $1 \leqslant i \leqslant (k-1)/2$, and $h(T_{-i}) = h(T_i) = H(1) = 0$ for $(k+1)/2 \leqslant i \leqslant b-1$, and $h(T_{-b}) = h(T_b) = H(1 + (t+1)/2) = 1 + (t+1)/2$. It is easy to check that $h(T) = H(n) + 1$ by virtue of Eqs. (6) and (7). This finishes the proof of Theorem 10 for $L = 1$.

For $L \geqslant 2$, we construct the desired CM-admissible tree T satisfying Eq. (14) by the following strategies:

For integer $n \geqslant 2$ with $(2b+1)^L \leqslant n < (2b+1)^{L+1}$. Let $n = (2b+1)^L + 2bk + t$, $0 \leqslant k < (2b+1)^L$, $0 \leqslant t < 2b$. Suppose that $A_1 : B_1 \mid A_2 : B_2 \mid \cdots \mid A_b : B_b$ is the first weighing of tree T.

- If $0 \leqslant k < (2b+1)^{L-1}$, $|A_i|$ and $|B_i|$ are given by Eq. (15);
- If $(2b+1)^{L-1} \leqslant k < (2b+1)^L$, we represent $k = m(2b+1)^{L-1} + q$ for some $1 \leqslant m < 2b+1$ and $0 \leqslant q < (2b+1)^{L-1}$, $|A_i|$ and $|B_i|$ are given by Eq. (16).

$$|A_i| = |B_i| = (2b+1)^{L-1} \quad \text{for } 1 \leqslant i \leqslant b, \tag{15}$$

$$|A_i| = |B_i| = \begin{cases} (2b+1)^L & \text{if } 1 \leqslant i \leqslant \left\lfloor \frac{m}{2} \right\rfloor, \\ (2b+1)^{L-1} & \text{if } \left\lfloor \frac{m}{2} \right\rfloor + 1 \leqslant i \leqslant b-1, \\ (2b+1)^{L-1} + 2b \cdot \left\lfloor \frac{(2b+1)^{L-1}+q}{2} \right\rfloor \cdot \left(m - 2\left\lfloor \frac{m}{2} \right\rfloor\right) & \text{if } i = b. \end{cases} \tag{16}$$

It is easy to check that if $0 \leqslant k < (2b+1)^{L-1}$ then $|C_0| = (2b+1)^{L-1} + 2bk + t$, and if $(2b+1)^{L-1} \leqslant k < (2b+1)^L$ then

$$\begin{aligned} |C_0| &= n - \sum_{i=1}^{b} (|A_i| + |B_i|) \\ &= (2b+1)^{L-1} + 2bq + t + 2b\left(m - 2\left\lfloor \frac{m}{2} \right\rfloor\right)\left\{(2b+1)^{L-1} - 2\left\lfloor \frac{(2b+1)^{L-1}+q}{2} \right\rfloor\right\}. \end{aligned}$$

In order to prove Theorem 10, we have to distinguish two cases (A) (k or t is even) and (B) (both k and t are odd).

(A) *k or t is even*.

(A.1) $0 \leqslant k < (2b+1)^{L-1}$. In this case, we have $|A_i| = |B_i| = (2b+1)^{L-1}$ for $1 \leqslant i \leqslant b$, and $|C_0| = (2b+1)^{L-1} + 2bk + t$. Lemma 8 shows that there exist CM-admissible trees T_{-i} and T_i such that

$$h(T_{-i}) = h(T_i) = H((2b+1)^{L-1}) = (2b+1)^{L-1} \times (L-1) \quad \text{for } 1 \leqslant i \leqslant b. \tag{17}$$

The induction hypothesis implies that there exists a CM-admissible tree T_0 such that $h(T_0) = H((2b+1)^{L-1} + 2bk + t)$. It is easy to check that $h(T) = H(n)$ by virtue of Eqs. (6) and (7).

(A.2) $(2b+1)^{L-1} \leqslant k < (2b+1)^L$. If m is even, we have $|A_i| = |B_i| = (2b+1)^L$ for $1 \leqslant i \leqslant m/2$, and $|A_i| = |B_i| = (2b+1)^{L-1}$ for $(m/2)+1 \leqslant i \leqslant b$, and $|C_0| = (2b+1)^{L-1} + 2bq + t$. Lemma 8 shows that there exist CM-admissible trees T_{-i} and T_i such that

$$h(T_{-i}) = h(T_i) = (2b+1)^L \times L \quad \text{for } 1 \leqslant i \leqslant m/2, \tag{18}$$

and

$$h(T_{-i}) = h(T_i) = (2b+1)^{L-1} \times (L-1) \quad \text{for } (m+2)/2 \leqslant i \leqslant b. \tag{19}$$

The induction hypothesis implies that there exists a CM-admissible tree T_0 such that $h(T_0) = H(|C_0|) = H((2b+1)^{L-1} + 2bq + t)$, as $0 \leqslant q < (2b+1)^{L-1}$, $0 \leqslant t < 2b$, and q or t is even (if t is odd, then k is even, so q must be even as $k = m(2b+1)^{L-1} + q$). Finally, $h(T) = H(n)$ is obtained by Eqs. (6) and (7).

If m is odd and q is even, we have $|A_i| = |B_i| = (2b+1)^L$ for $1 \leqslant i \leqslant (m-1)/2$, and $|A_i| = |B_i| = (2b+1)^{L-1}$ for $(m+1)/2 \leqslant i \leqslant b-1$. Lemma 8 shows that there exist CM-admissible trees T_{-i} and T_i such that

$$h(T_{-i}) = h(T_i) = (2b+1)^L \times L \quad \text{for } 1 \leqslant i \leqslant (m-1)/2, \tag{20}$$

and

$$h(T_{-i}) = h(T_i) = (2b+1)^{L-1} \times (L-1) \quad \text{for } (m+1)/2 \leqslant i \leqslant b-1. \tag{21}$$

We also have $|A_{-b}| = |B_b| = (2b+1)^{L-1} + 2bk' + t'$, where $k' = ((2b+1)^{L-1} + q - 1)/2$, $t' = 0$; and $|C_0| = (2b+1)^{L-1} + 2bk'' + t''$, where $k'' = 1$, $t'' = t$. Now, k is odd as $k = m(2b+1)^{L-1} + q$ (m odd and q even), so t must be even in the present case. In both cases, $0 \leqslant k', k'' < (2b+1)^{L-1}$, $0 \leqslant t', t'' < 2b$, and both t' and $t'' = t$ are even. The induction hypothesis implies that there exist CM-admissible trees T_0, T_{-b} and T_b such that

$$h(T_0) = H(|C_0|) = H((2b+1)^{L-1} + 2b + t)$$

and

$$h(T_{-b}) = h(T_b) = H((2b+1)^{L-1} + 2bk'). \tag{22}$$

Finally, $h(T) = H(n)$ is obtained by Eqs. (6) and (7).

If both m and q are odd, we have $|A_i| = |B_i| = (2b+1)^L$ for $1 \leqslant i \leqslant (m-1)/2$; and $|A_i| = |B_i| = (2b+1)^{L-1}$ for $(m+1)/2 \leqslant i \leqslant b-1$; and $|A_{-b}| = |B_b| = (2b+1)^{L-1} + 2bk' + t'$, where $k' = ((2b+1)^{L-1} + q)/2$, $t' = 0$; and $|C_0| = (2b+1)^{L-1} + 2bk'' + t''$, where $k'' = 0$, $t'' = t$. In both cases, $0 \leqslant k', k'' < (2b+1)^{L-1}$, $0 \leqslant t', t'' < 2b$, and both k'' and t' are even. Similarly, the induction hypothesis, Lemma 8, Eqs. (6) and (7) give $h(T) = H(n)$.

(B) *Both k and t are odd.* In this case, we cannot get any CM-admissible tree which has external path length $H(n)$. Suppose, on the contrary, that there exists a CM-admissible tree T with $h(T) = H(n)$, i.e., T is a Huffman tree. On the one hand, T has $n - \lceil (2bk+t)(2b+1)/2b \rceil = (2b+1)^L - k - 1$ leaves at level L of T (by letting $q = 2b$ in Lemma 2). On the other hand, both k and t being odd implies that $n = (2b+1)^L + 2bk + t$ is even. Thus Lemma 9 shows that the number of leaves at level L of T is even. This is a contradiction to the fact that $(2b+1)^L - k - 1$ is odd. Therefore, we can conclude that $h(T) \geqslant H(n) + 1$ for any CM-admissible tree T when both k and t are odd.

We now prove that the strategies defined by Eqs. (15) and (16) give a CM-admissible tree T with $h(T) = H(n) + 1$.

For $0 < k < (2b+1)^{L-1}$, we have $|C_0| = (2b+1)^{L-1} + 2bk + t$. The induction hypothesis (both k and t are odd) implies that there exists a CM-admissible tree T_0 such that

$$h(T_0) = H((2b+1)^{L-1} + 2bk + t) + 1.$$

This equality, together with Eqs. (17), (6) and (7), gives $h(T) = H(n) + 1$.

For $(2b+1)^{L-1} \leqslant k < (2b+1)^L$ and m even, we have $|C_0| = (2b+1)^{L-1} + 2bq + t$. Note that k is odd and m is even in the equality $k = m(2b+1)^{L-1} + q$, so both q and t are odd ($0 \leqslant q < (2b+1)^{L-1}$) in the present case. The induction hypothesis implies that there exists a CM-admissible tree T_0 such that

$$h(T_0) = H((2b+1)^{L-1} + 2bq + t) + 1.$$

This equality, together with Eqs. (18), (19), (6) and (7), gives $h(T) = H(n) + 1$.

For $(2b+1)^{L-1} \leqslant k < (2b+1)^L$ and m odd, the equality $k = m(2b+1)^{L-1} + q$ tells us that q is even. So we have $|A_{-b}| = |B_b| = (2b+1)^{L-1} + 2b \cdot ((2b+1)^{L-1} + q - 1)/2 = (2b+1)^{L-1} + 2bk' + t'$, where $k' = ((2b+1)^{L-1} + q - 1)/2$, and $t' = 0$. Thus, Eq. (22) is true for the present case. We also have $|C_0| = (2b+1)^{L-1} + 2b + t = (2b+1)^{L-1} + 2bk'' + t''$, where $k'' = 1$ and $t'' = t$ are odd. The induction hypothesis ($k'' = 1 < (2b+1)^{L-1}$ as $L \geqslant 2$, $t'' = t < 2b$) implies that there exists a CM-admissible tree T_0 such that

$$h(T_0) = H((2b+1)^{L-1} + 2b + t) + 1.$$

This equality, together with Eqs. (20), (21)–(22), (6) and (7), gives $h(T) = H(n) + 1$. The proof of Theorem 10 is complete. □

Corollary 11. *Let $n = (2b+1)^L + 2bk + t \geqslant 2$ for some $0 \leqslant k < (2b+1)^L$ and $0 \leqslant t < 2b$. Suppose that T^* is the CM-admissible tree constructed in Theorem 10 with $\bar{L}_{CM}(n) = h(T^*)/n$. Then T^* has m leaves at level L and $n - m$ leaves at level $L + 1$ for some integer m, i.e., all leaves of T^* are at level L or $L + 1$ of T^*.*

Proof. It can be easily verified by the proof of Theorem 10. □

6. Conclusions

One of the well-known search problems is how to find a heavier coin among a set of n coins when the test device is $b = 1$ balance (two-arms balance). Let $L^{(1)}(n)$, $L^{(1)}_{\mathrm{pre}}(n)$, $\bar{L}^{(1)}(n)$ and $\bar{L}^{(1)}_{\mathrm{pre}}(n)$ be the minimum number of weighings for the worst-case sequential algorithm, the worst-case predetermined algorithm, the average-case sequential algorithm and the average-case predetermined algorithm, respectively. Aigner [1, Theorem 2.1] showed that $L^{(1)}_{\mathrm{pre}}(n) = L^{(1)}(n) = \lceil \log_3 n \rceil$. By letting $b = 1$ in Theorem 1 of this paper, $L^{(1)}_{\mathrm{pre}}(n)$ and $L^{(1)}(n)$ can be determined, which are the same with Aigner's results. By letting $b = 1$ in Theorems 3 and 6 of this paper, $\bar{L}^{(1)}_{\mathrm{pre}}(n)$ and $\bar{L}^{(1)}(n)$ can be obtained, respectively. It is easy to see that they coincide with the results of Aigner [1, Theorem 2.2].

Aigner [1] has studied the **CM** in the case that the test device is $b = 1$ balance, and has given the following result: For $3^L \leqslant n < 3^{L+1}$, the equality $\bar{L}_{CM}(n) = H(n)/n$ cannot be achieved if $3^{\lceil \log_3 n \rceil} - n \equiv 3 \pmod 4$, in the other cases it is correct. But the expression on $\bar{L}_{CM}(n)$ is not given if $3^{\lceil \log_3 n \rceil} - n \equiv 3 \pmod 4$. Theorem 10 provides the explicit expression on $\bar{L}_{\mathrm{CM}}(n)$ when the test device is b-balance. In fact, it can be proved that $3^{\lceil \log_3 n \rceil} - n \equiv 3 \pmod 4$ is equivalent to the condition in Theorem 10 when $\bar{L}_{\mathrm{CM}}(n) = (H(n)+1)/n$ and $b = 1$. Therefore, Theorem 10 has generalized the above result of Aigner.

It would be interesting to investigate the present models in presence of lies. Pelc [15], Liu [10] give the minimum number of weighings for the worst-case sequential algorithm when $b = 1$ and one lie, $b = 1$ and two lies, respectively. For more results on the problems of searching with lies, we refer the reader to the survey paper [16].

Acknowledgements

This work was supported by the National Natural Science Foundations of China, Grant Nos. 10571045 and 70571024. The authors would like to thank the Editor-in Chief, Editor and the anonymous referees for their useful suggestions and comments on the earlier version of the paper.

References

[1] M. Aigner, Combinatorial Search, Wiley-Teuber, New York, 1988.
[2] A. De Bonis, A predetermined algorithm for detecting a counterfeit coin with a multi-arms balance, Discrete Appl. Math. 86 (1998) 181–200.
[3] A. De Bonis, L. Gargano, U. Vaccaro, Optimal detection of a counterfeit coin with multi-arms balances, Discrete Appl. Math. 61 (1995) 121–131.
[4] L. Gargano, J. Körner, U. Vaccaro, Search problems for two irregular coins with incomplete feedback: the underweight model, Discrete Appl. Math. 36 (1992) 191–197.
[5] L. Halbeisen, N. Hungerbühler, The general counterfeit coin problem, Discrete Math. 147 (1995) 139–150.
[6] F.K. Hwang, A tale of two coins, Amer. Math. Monthly 94 (1987) 121–129.
[7] G.O.H. Katona, Combinatorial search problems, in: J.N. Srivastava, et al. (Eds.), A Survey of Combinatorial Theory, North-Holland, Amsterdam, pp. 285–308.
[8] N. Linial, M. Tarsi, The counterfeit coin problem revisited, SIAM J. Comput. 11 (1982) 409–415.
[9] W.A. Liu, Z.K. Nie, Optimal detection of two counterfeit coins with two-arms balances, Discrete Appl. Math. 137 (2004) 267–291.
[10] W.A. Liu, Q.M. Zhang, Z.K. Nie, Searching for a counterfeit coins with two unreliable weighings, Discrete Appl. Math. 150 (2005) 160–181.
[11] W.A. Liu, Q.M. Zhang, Z.K. Nie, Optimal search procedure on coin-weighing problem, J. Statist. Plann. Inference, to appear.
[12] W.A. Liu, W.G. Zhang, Z.K. Nie, Searching for two counterfeit coins with two-arms balance, Discrete Appl. Math. 152 (2005) 187–212.
[13] A.J. Macula, A non-adaptive search algorithm that identifies up to three defects, J. Statist. Plann. Inference 60 (1997) 269–272.
[14] A.J. Macula, G.R. Reuter, Simplified searching for two defects, J. Statist. Plann. Inference 66 (1998) 77–82.
[15] A. Pelc, Detecting a counterfeit coin with unreliable weighings, Ars Combin. 27 (1989) 181–192.
[16] A. Pelc, Searching games with errors—fifty years of coping with liars, Theoret. Comput. Sci. 270 (2002) 71–109.
[17] R. Tǒsić, Tow counterfeit coins, Discrete Math. 46 (1983) 289–295.
[18] R. Tǒsić, Five counterfeit coins, J. Statist. Plann. Inference 22 (1989) 197–202.